e-Prime - Advances in Electrical Engineering, Electronics and Energy 6 (2023) 100332

Contents lists available at ScienceDirect

e-Prime - Advances in Electrical Engineering, Electronics and Energy

journal homepage: www.elsevier.com/locate/prime

An enhanced approach towards improving the performance of embedding memory management units into Network-on-Chip

Debasis Behera [a,b], Suvendu Narayan Mishra [b], Prabodh Kumar Sahoo [*,c], Heli Amit Shah [c]

[a] Department of Electrical and Electronics Engineering, C. V. Raman Global University, Bhubaneswar, Odisha, India, 752054
[b] Department of Electronics and Telecommunication Engineering, Veer Surendra Sai University of Technology, Burla, Odisha, 768018
[c] Department of Mechatronics, Parul Institute of Technology, Parul University, Waghodia, Vadodara, Gujarat, India, 391760

ARTICLE INFO

Keywords:
FIFO
Memory Management Unit (MMU)
NoC
Multi processor system
Distribute MMU
Centralized MMU

ABSTRACT

Network-on-Chip (NoC) initiates the design procedure of interconnection network into SoC - System-on-Chip. The current technique overcomes the drawbacks of traditional bus-based SoC, for instance, poor scalability, small-link bandwidth, and large delay. The network-on-Chip systems required minimum buffering space and low-latency requirements. Here, the FIFO buffer is utilized as a virtual channel to evade deadlock issues and thus enhances throughput. The current research focused on an enhanced toward improving Embedding-Memory-management- Units performance into Network-on-Chip. Here, the 3D NoC was considered in the study for Noc improvement and has achieved high performance. Further, A first-in-first-out (FIFO) buffer was used in NoC routers to store the data packets temporarily. Eventually, the RAM -random access memory was proposed that serves as a link between the crossbar switch and the input ports. The simulation result of the current research results in only 0 to 16 memory for data storage in a stack out of 64, with the almost empty signal being high.

Introduction

The multi-processor cores usage has been rapidly increasing; therefore, the inherent flaws in data communication and memory access are revealed. An advanced framework called "NoC" - network-on-chip has been developed with the chip communication of heterogeneous systems in embedded applications and multi-core systems to solve these inherent flaws. Compared to the traditional bus system, the NoC has considerably exhibited higher performance, and it is considered a new on-chip communication method [1,2],. The NOC applications that are based on communications have become a new flashpoint in the design of integrated circuits. The manufacturers have gradually increased the production and investment in this specific field. Hence, the NoC-based framework [3] will definitely create a great development and impact on the multi-core distribution system. Compared to the other systems like distributed and centralized, NoC has the potential benefit of dissipating workload to decentralized control, scalability, multi-core system, concurrent operation, core system, and heterogeneous subsystem.

Further, multi-core communication is considered as the powerful core technologies, especially in distributed-systems. The efficiency of inner-core communication is the vital index that may affect the distributed-multi-core system performance, and also it helps in data transferring. The CMP - Chip Multiprocessor is a future generation design architecture that comprises thousands of cores to enhance system performance. Therefore, NoC is considered a new design for the intercommunication of the SoC - System on Chip. To achieve greater performance, 3-D NoC has been designed from NoC fabric. However, the main limitation of NoC is due to the memory unit and symmetrically boosts in 3D NoC. Normally, in many existing systems, FIFO - first-in-first-out [4] has been positioned in the router of NoC in order to store data-packet tentatively. Here, memory is considered the main problem for NoC-based SoS due to the cost and size of the device. Various advanced architectures on memory like RAM - Random Access Memory [5] have been employed to optimize the NoC router performance and processed as a bridge amidst the crossbar switch and ports (input). This strategy's limitations in terms of flexibility and scalability, as well as the complexity of the design process, are drawbacks. Time closure and quality of service are increasingly difficult to achieve the more cores connected to the bus. The memory unit is one of the main shortcomings of NoC, and it gets worse in 3-D NoC.

Moreover, maintaining the privacy and integrity of the apps plays a significant role in mobile devices and data is considered of extreme importance. This normally includes the SoC that incorporates peripheral IP - Intellectual Property and microprocessors that an NoC links. The

* Corresponding author.
E-mail address: prabodh.sahoo29574@paruluniversity.ac.in (P.K. Sahoo).

https://doi.org/10.1016/j.prime.2023.100332
Received 9 August 2023; Received in revised form 27 September 2023; Accepted 18 October 2023
Available online 24 October 2023

various malicious attacks may target the critical data, and NoC utilizing MMUs - Memory Management Units or other mechanisms have the possibility to manage the data transfers and block the suspicious attacks. Nevertheless, commodity processors will not offer persistent assurances based on the accurateness of the mechanisms. Certainly, it is difficult to identify the entire access control mechanisms in the system and whether it is properly configured. Hence, the NOC firewall [6] has been utilized in the research that states locus control, and it is flexible for formal analysis.

The PE - Processing Elements incorporated in MPSoC - Multiprocessor System-on-Chip and NoC are examined as an optimistic scheme in the recent studies. Therefore, the chip network can procure reliable, low-latency, high-bandwidth, and scalable communication. At the same time, NoC development on large memory includes shared caches and private that help in parallel transactions along with multiple cores. The memory subsystem that captures the power and NoC area, therefore, the MPSoC context has been classified into two memory schemes: centralized shared-memory and distributed-memory framework (multi-processor) [7]. Nevertheless, delivering the required NOC bandwidth and low-access latency is complex.

Furthermore, the AMM - Adaptive Memory Management [8] methods have been utilized to manage the efficiency. Hence, the performance of the MMU-less embedded systems with the appropriate timing behavior by increasing the embedded system [9] and memory utilization of the applications for performance optimization during time-critical tasks and fulfilling the IoT demands without failures and memory leaks. Further, it is significant to find the problems that include lack-of-hardware features [10], memory constraints, injection attacks [11], data flow hijack [12], memory corruptions [13], and Mirai (Infamous attacks) [14] and many more in the memory system. It is vital to enhance the securities in the embedded system. Certainly, it is difficult to design an NoC that covers all the performance metrics, including signal integrity, power consumption, latency, and system scalability. However, the existing techniques developed an advanced memory structure in order to acquire effective data transfer with the FIFO buffers that are applied in virtual channels and distributed RAM for FPGA- Field programmable gate array-based NoC. Further, the performance has been estimated with BiNoC - Bi-directional NoC [15].

In this research, we focus on an enhanced approach toward improving the performance of Embedding-Memory-Management-Units into Network-on-Chip. 3D NoC is indicated as the unique way of enhancement in NoC fabric to obtain high performance. A first-in-first-out (FIFO) buffer is commonly used in NoC routers that helps in data packets storage, temporarily. Because of the device's size and cost, memory is a key concern for NoC-based SoCs. To improve the performance of the NoC router, an advanced memory architecture is described. The random access memory (RAM) is proposed, which serves as a link between the input ports and the crossbar switch.

Literature review

In [16], an enhanced fault tolerance resource allocation method has been implemented on multicore platforms that provide an energy-aware resource allocation in different scenarios like persistent core failures. Here, the scheduling approach and unified mapping have been utilized in order to find out the faulty and fault-free scenarios. The algorithm utilized in this study has procured better results in terms of quality of service, communication energy, and reconfiguration overhead, and it was compared to the other fault-tolerance techniques.

In [17], a GPU-based NoC simulation technique has been developed to deliver easy extensibility, thread safety, and high fidelity in hardware. Therefore, these properties are relevant for the simulation of parallel NoC in the entire units, including GPU. The presented module-group abstraction that decouples the modules definition taken from the mapping on the specific platform has resulted in its ability to process the simulation in large networks with ~4 K nodes on GPU and the acquired speedups over 250x. However, still, challenges are stated, including memory usage, synchronization issues, and thread divergence. Further, it is necessary to validate and model appropriate NoC designs with thousands of nodes in future research.

In [18], a reconfigurable cluster-based NoC framework has been implemented for NN - Neural Networks. Therefore, this architecture normally, endorsed by a flow diagram that connects the NN onto the NoC cluster, further defines a topology on the basis of the reconfigurable inter-cluster connectivity model of the network. The experimental results procured 71 % - higher throughput compared to a conventional mesh while expediting the neural processing.

In [19], the MMNoC design includes MMU - Memory Management Unit functionality, and NoC has been utilized in the embedded system, pointing to low cost and low power. This enables the multiprocessing units in the system platforms; these designs were incorporated into NoC with certain AsNoC - application-specific NoC designs. There is no necessity to do modifications to the NoC architecture, and this MMNoC has the possibility to use in different NoC types. Further, MMNoC enables the hardware engineers in the embedded system to improve the lightweight processors that do not include MMU to create a platform that benefits multiprocessing. Eventually, the experimental results showed that the capability of multiprocessing might surpass the associated overhead.

In [12], the Control-Flow Attestation - C-FLAT has developed an specific application operating on an embedded machines. Therefore, C-FLAT correlates static-attestation by evaluating the binary level execution path and obtaining the runtime behavior. Further, C-FLAT defines a significant and considerable advance concerning addressing the open issues of runtime attestation and enables the prover to expeditiously calculate the program's control flow through aggregate authentication with the proper instructions consisting of function returns and branches. Therefore, this specific authenticator in the process indicates a fingerprint signature of the control-flow path that enables the verifier to track the correct path during execution to define and understand the control flow of the applications that are compromised. Eventually, incorporating C-FLAT with static attestation can accurately attest to the execution in embedded software that allows identifying the runtime attacks.

In [20], a software-based approach has been utilized in order to restore the deterministic memory in the embedded system with high performance. The utilized technique helps enhance the transparency of the OS - Operating System and DMA-Friendly cache coloring incorporation with IDA - invalidation-driven allocation method. Certainly, this approach allows for securing essential cache blocks procured from internal and external eviction. Further, the experimental results obtained by prolonging the Jailhouse partitioning hypervisor were employed and estimated with the amalgamation of real and synthetic benchmarks. These techniques are implemented with memory coloring to execute DRAM - Dynamic Random Access Memory and cache by extending the virtualization in the modern embedded system.

Proposed methodology

This research mainly concentrated on an improved method for enhancing the Embedding- Memory-Management-Units' performance in Network-on-Chip. Obtaining high-performance 3D-NoC is the unique process of enhancement in NoC-fabric. FIFO buffer is normally utilized in NoC routers in data packets storage, temporarily. Due to the device's size and cost, memory is considered a key concern for NoC-based SoCs. To enhance the NoC router performance, an advanced and optimized memory architecture is described; hence the random access memory (RAM) is proposed, which serves as a link between the input ports and the crossbar switch.

MMU embedded NoC

One of the most optimal solutions for intercommunication concerns is Network on Chip (NoC). As technology advances, several advancements are seen in the NoC architecture, which is 3-D NoC. PEs - Processing Elements are inter-connected horizontally and vertically in a 3D-NoC, resulting in improved performance. In the event of high traffic, PEs can transmit data among themselves, lowering latency and power consumption [5].

Fig. 1. defines the MMU for the PEs present in the SoC design that is considered a shared memory unit. Therefore, this is utilized to transfer the data in PE, which is for general point in the FPGA-based-multiprocessor-system. The presented shared memory system framework is also denoted as CoRAM - Connected-RAM that has the potential to reconfigure both heterogeneous fabric and standalone. This reconfigure logic is taken from the provided memory storage via the memory interface system present in NoC along with the linear address boundaries [5].

System architecture

Here propose, the distributed memory units that display the centralized MMU structure have been depicted in Fig. 2, which connects all PEs to centralized memory. The difficulty with centralized-MMU with maximum network latency as a result of multiple requests striking the MM. Additionally, the on-chip-network that connects the PE to the centralized memory is another key issue, causing scaling challenges. This study presents the distributed-MMU, that needs a larger count of MMUs, to avoid certain bottlenecks that helps in memory access requests management presented in NoC. Enhanced NoC is demonstrated by high-bandwidth usage and minimum traffic with optimal MMU placement [5].

The PE uses a distributed MMU depicted in 3. To request the primary memory channel, which is based on specific routing tables in the research. Every clock cycle, these represented routing tables are completely refreshed, assigning requests to the idle MMU. Therefore, the current router is in charge of memory requests along with its distributed MMU, allowing it to access the main memory if available, otherwise delegating it to another MMU. Each distributed MMU is assigned to a certain PE's memory request, and if that MMU is unavailable or overloaded, it is assigned to dedicate MMU (other). During the execution, the MMU receives a request, especially for accessing memory, and it decodes the specific address and presents access based on the availability, injecting data-packets into the certain network executed in the process. The almost-empty/almost-full flag bit identifies read and writes operations utilzing serially coupled T-flip flop (TFF) circuitry.

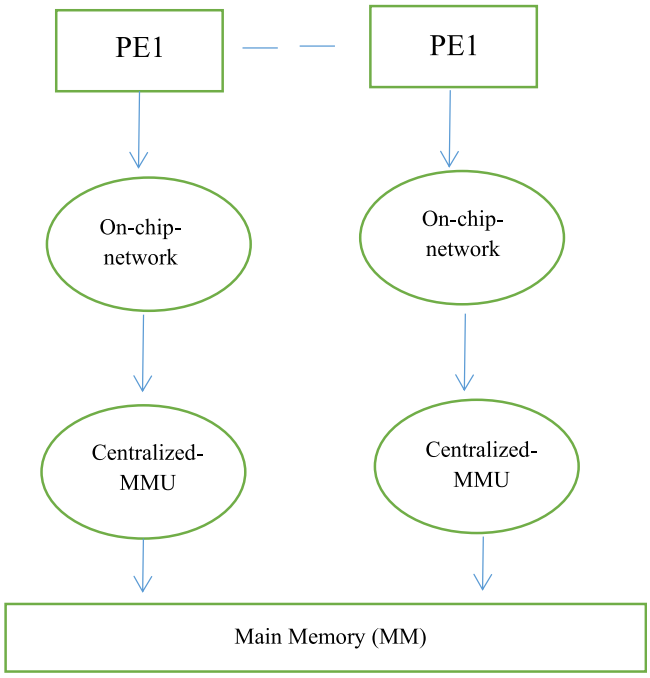

Fig. 2. Centralized MMU.

Both operations such as write and read are performed with the help of an address-pointer that two serially coupled TFFs control, which has been illustrated in 4. The TFFs are XNORed for read-write operations and started in '10′ mode. The empty flag is set to high during the values of the serially interconnected TFFs and address pointer are equal. Hence the value of the specific address pointer is equal, but the TFFs are not, indicated with the full flag asserts high. The proposed pattern can be analyzed in usage of area, delay, memory, throughput, and power consumption while implemented on FPGA devices. Because of its versatility and ease of prototyping, the FPGA is employed as the end-device for VLSI-technology.

Empty/Full flag behavior for FIFO in MMU

FIFO controller

➢ The FIFO-controller, certainly deasserts the EMPTY-flag for 2clock cycles that are read immediately once the deassertion of the CLR signal was done while the read_clock (RCLK) pulse is maximum.

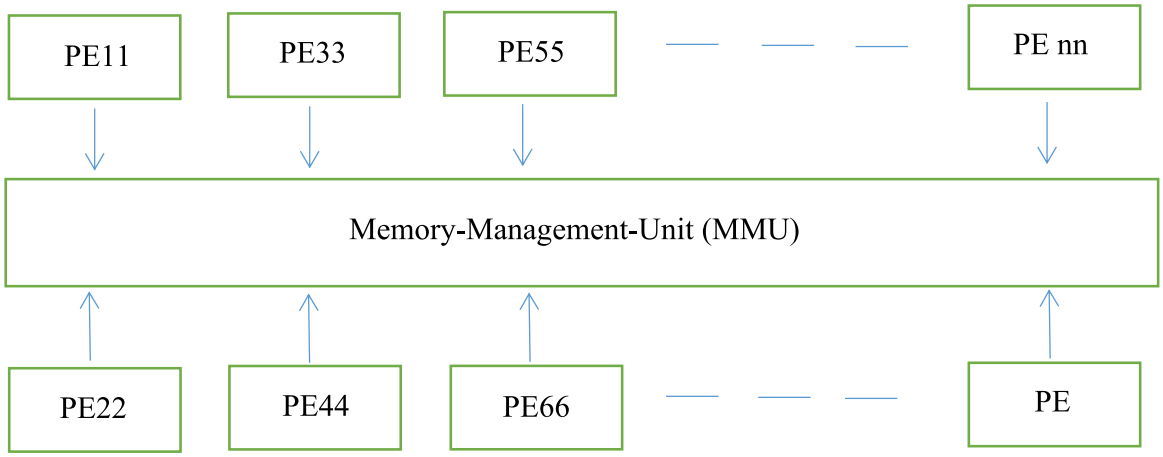

Fig. 1. FPGA-based Multiprocessor Framework with Shared Memory [5].

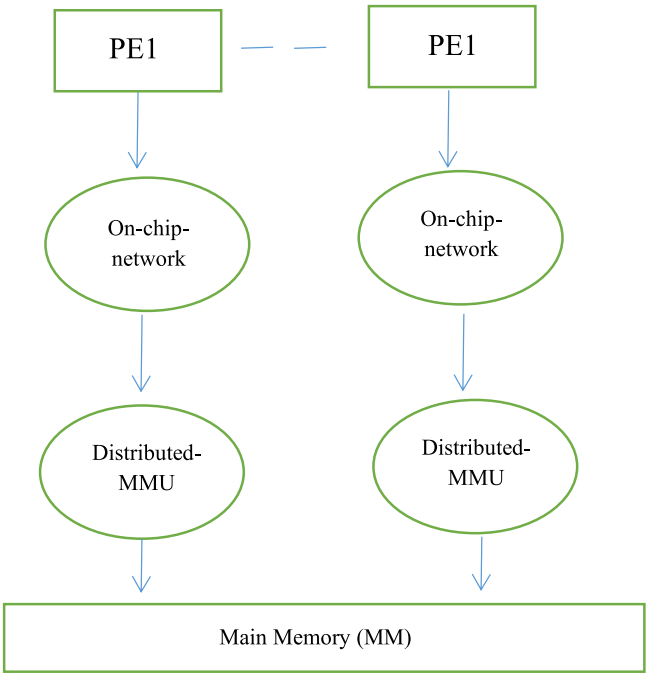

Fig. 3. Distributed MMU.

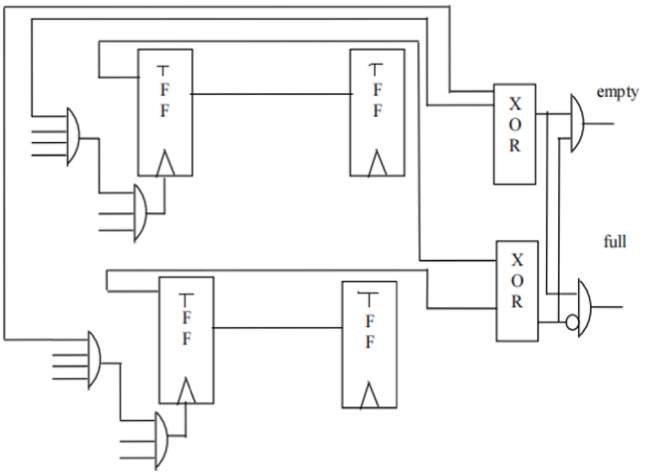

Fig. 4. Two serially coupled TFFs Circuit Diagram.

➢ The FIFO-controller could have the possibility to falsely assert the FULL-flags and EMPTY- flags if separate clock frequencies are utilized for WCLK and RCLK.

The reason for utilizing almost-full and almost-empty is to deliver an 'advance' warning to the remaining system that the FIFO limits are attained, and the FIFO controller has been depicted in Fig. 5. Therefore, this provides the system enough time to avoid and react under-run over- or conditions. For instance, a device enabling a FIFO has the possibility to take more than one in order to answer a real indication so that it can be overrun. Further, by providing a timely alert with almost full, the host can notice it in time to stall its writing and evade FIFO overrun. A skid buffer is used to find the difference between full forms and almost full. The main utilization is almost-empty for the FIFO writer to understand the communication of sending new data before the reader completely drains the FIFO.

Further, the warning enables the writer enough time; in this instance, it is significant to set up the upcoming transfer in order to push the new data into the FIFO. Therefore, it has been utilized for a data sink without any tolerance for this tall, including DAC - Digital to Analog Converter that required continuous data. The almost-empty use-cases balance for reading the latency response to the empty flag. Hence, utilizing almost-empty provides more time for sampling the flag and stall the provided reading in a specific period in order to evade reading or analyze the FIFO to under-run. These are related to the other cases read. Normally, the reader will carry one cycle in order to respond to not-empty. Due to its sampling processing delay and persistence in an earlier stage, the reading will be processed 'beyond the end' while the FIFO is emptied, also known as under-run. So it is significant to come up with a dumb solution, where the 'dead' cycle can be inserted amidst each reading, and these will be executed with an empty flag sampling before beginning the read.

Moreover, various protocols include AXI-Stream, which deals with the different scenarios by qualifying data and the proper valid flag. Certainly, Not-empty is utilized as 'valid' in the read case of FIFO, where the readers' waiting period will be completely eradicated. On the contrary, it initiates the read whenever needed; however, it does not dispatch the data until the following conditions become true: the FIFO 'valid' which is not empty and 'ready.'

Results and discussions

This section has elaborated on the simulation findings regarding the almost empty/full flag behavior for FIFO in MMU. In this instance, the FIFO buffer utilized by Router Logic and NIC is built with an additional warning signal of almost_full, also known as "underflow," and almost_empty, sometimes known as "overflow," which serves as an "advance" warning to the other blocks that restrict the FIFO. The collected signals essentially stop the systems from operating in an over- or under-run state. As a result, the research has explained how the ALMOST-EMPTY and ALMOST-FULL signals are generated. To prevent writing to a FULL FIFO or reading from an empty FIFO, the data was transferred at high speed with the receipt of a FULL/EMPTY signal, and it was not executed and recognized in a certain time frame. Therefore, a particular process should be suitable to provide various advanced warnings under certain circumstances.

Additionally, signals for ALMOST FULL and EMPTY were used to provide the early warning. Similar to how signals such as EMPTY/FULL developed, signals such as ALMOST FULL/EMPTY did as well. Occasionally, during the ALMOST FULL signal, the results of the register write shift are different from the results of the processed shift register, which is similar to the shift register (read). But in those cases, the logical value was regarded as an offset from the real value displayed in the read shift register. The offset that read shift register had numerous logical values encoded in it. The ALMOST-EMPTY signal is the result of the read shift register rather than a programmed write-offset shift register at the same time.

By convention, during the affirmation of an ALMOST-FULL signal with the FULL (asserted) and the affirmation of the EMPTY signal with EMPTY (asserted). As denoted, ALMOST EMPTY and ALMOST FULL signals final generation consists of assertion in the scenario with the EMPTY or FULL signals, respectively. Certainly, the generation of the ALMOST FULL/ EMPTY signals is not restricted to utilizing FULL or EMPTY signal development. The top module of FIFO memory includes certain factors such as input data, output data, fifo_full, fifo_empty, fifo_underflow, fifo_overflow, wr(write), and rd(read) signal. The sub-modules contain certain information regarding the stack memory. This contains stack memory with a size of 64 to store data bits of 8-bit (data_in[7:0] with memory address for write and read output_data (data_out[7:0]) of 6-bit raddr [5:0] and waddr [5:0]. The pointer consists of certain details such as a read pointer, write enable (we), and write pointer. Eventually, the simulation results are exhibited in Figs. 6 and 7.

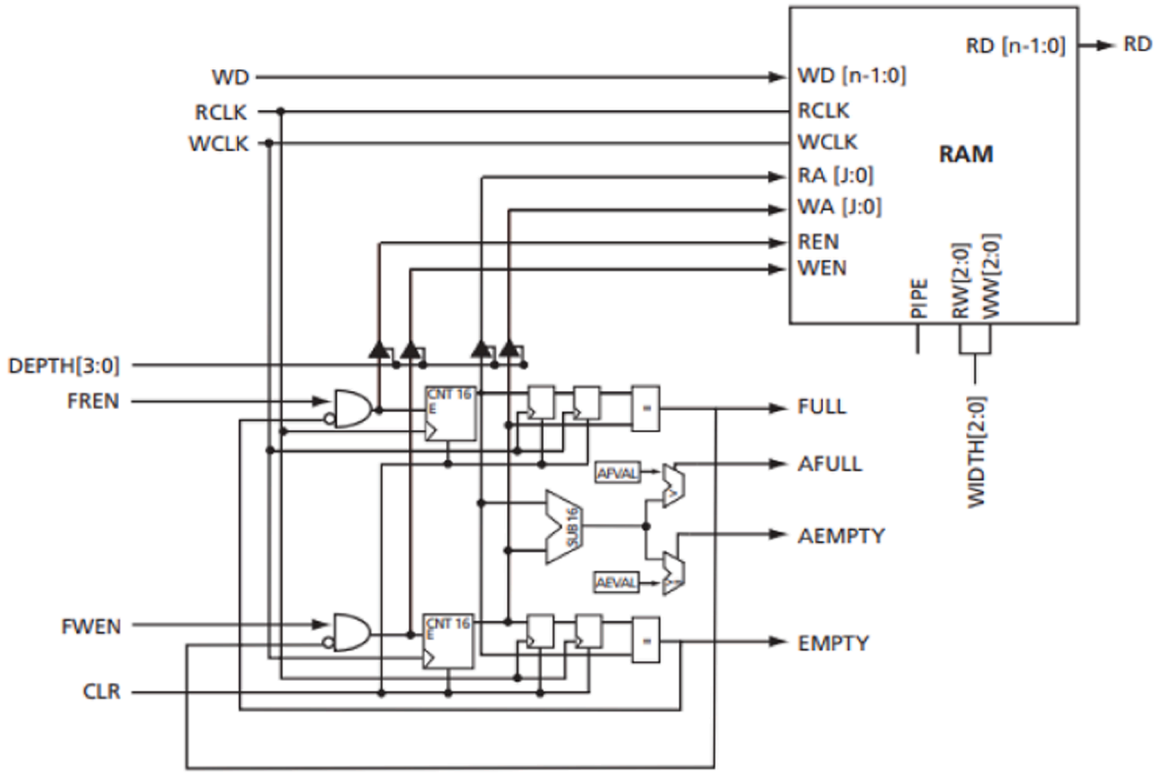

Fig. 5. FIFO Controller.

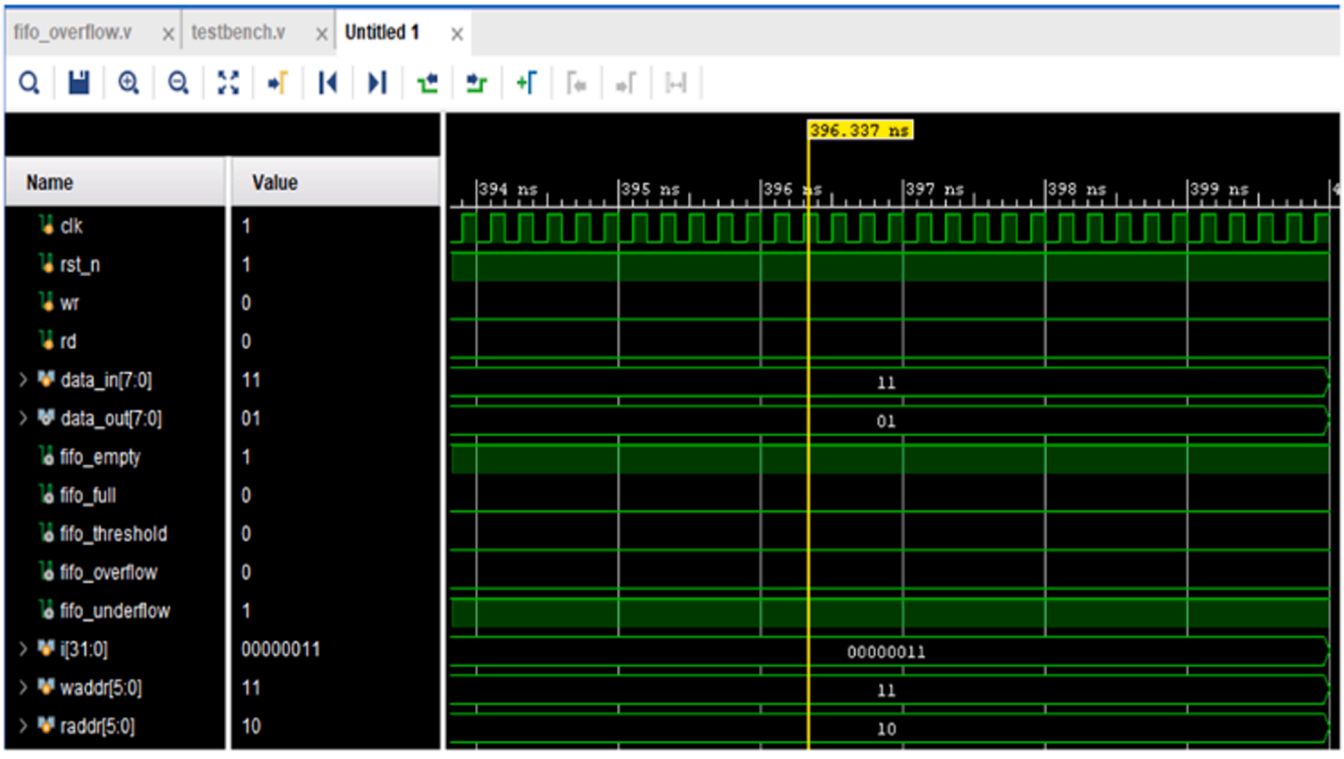

Fig. 6. Simulation Results on Almost Empty/Full Flag behavior for FIFO in MMU.

The outcomes of the distributed MMU-based NoC router simulation. The simulation verifies the empty/full flag with all possible combinations. The FIFO is initialized with an empty flag and asserts a full flag when it is filled. Because read operation requires more clock cycles for verification, write operation requires fewer clock cycles.

The current research concentrated on an improved method for enhancing the Embedding- Memory-Management-Units' performance in Network-on-Chip. Obtaining high-performance 3D NoC is the unique

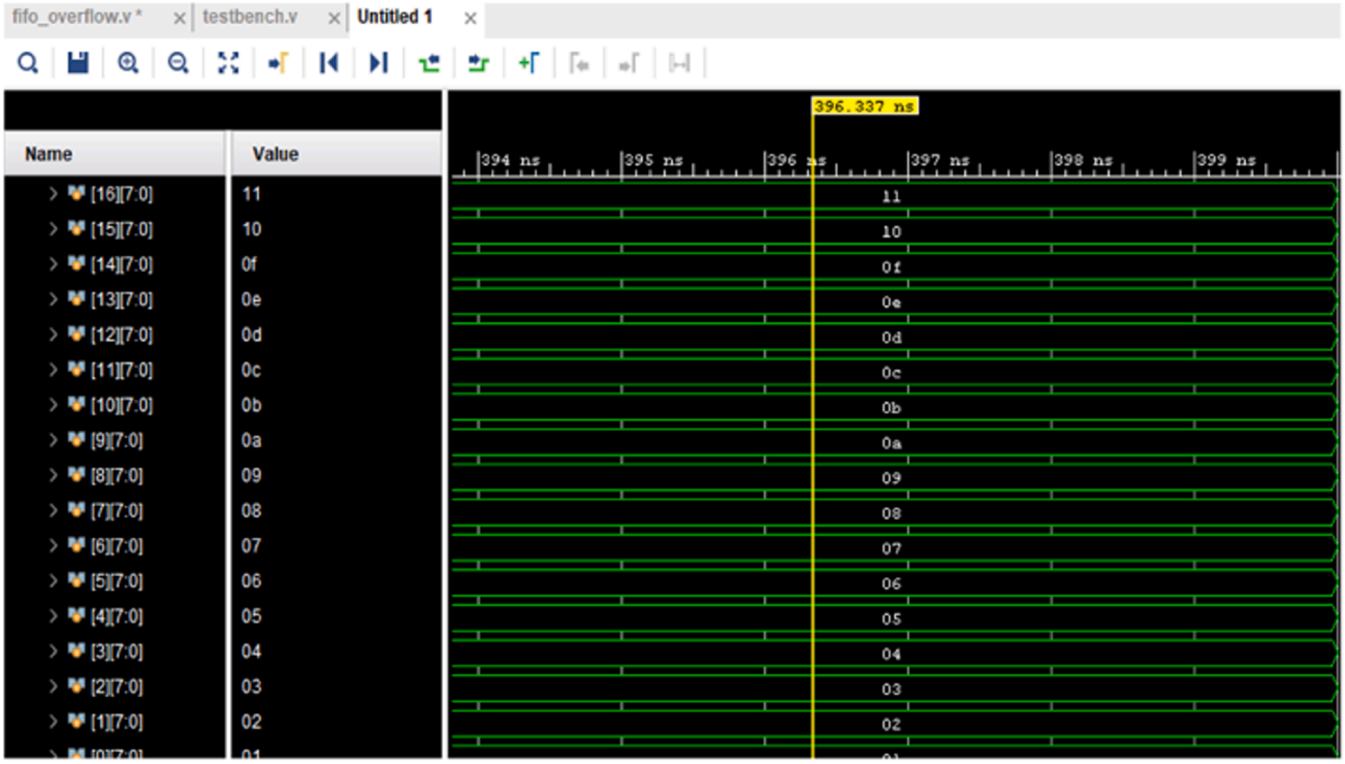

Fig. 7. Stack of Memory.

way of NoC fabric enhancement. The simulation result of the current research results in only 0 to 16 memory for data storage in a stack out of 64, with the almost empty signal being high.

Table 1 shows that the comparison of Resource utilization NoC has better performance than recent work due to the FIFO-based storage distributed MMU and the 3-D structure for the NoC. Although Bio-Noc has the same needs for the storage unit needed for this work, the operation frequency is better because of the empty/full flags that reduce the latency. 3 × 3 mesh-based NoC studies are repeated as many times as necessary to determine the precise performance in terms of throughput and delay Figure shows the 7 and 8.

Conclusion

The current research focused on an enhanced toward improving the Embedding Memory Management Units' performance in Network-on-Chip. The 3D NoC was considered the new way of NoC improvement that achieved high performance. A first-in, first-out (FIFO) buffer was commonly utilized in NoC routers to store data packets temporarily. Due to the cost and size of devices were considered a key concern for NoC-based SoCs. The NoC router performance has been improved with an advanced memory architecture. Eventually, the RAM -random access memory was proposed that serves as a link between the crossbar switch

Table 1
Comparison of resources utilization.

NOC	Resource utilized LUTs	Resource utilized FFs	Delay node (ns)
Proposed work	297	710	0.0246
A One-Cycle FIFO [4]	461	3957	0.015
Bi-NoC [15]	327	728	0.511
Distributed MMU based 3-D NOC [5]	390	3582	maximum delay of 3 × 3

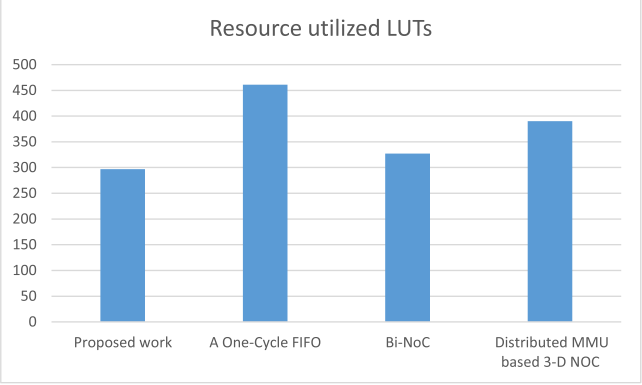

Fig. 8. Comparison of the proposed work with resource utilized LUTs.

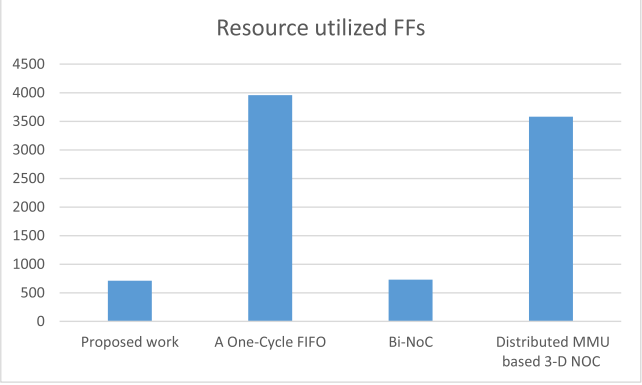

Fig. 9. Comparison of the proposed work with resource utilized FFs.

and the input ports. The simulation result of the current research results in only 0 to 16 memory for data storage in a stack out of 64, with the almost empty signal being high.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

Data will be made available on request.

References

[1] K. Zhang, Y. Chu, Design of distributed storage multi-core system based on NoC Architecture, in: 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), 2019, pp. 905–908. IEEE.

[2] A.K. Swain, A.K. Rajput, K.K. Mahapatra, Network on chip for consumer electronics devices: an architectural and performance exploration of synchronous and asynchronous network-on-chip-based systems, IEEE Consum. Electron. Mag. 8 (3) (2019) 50–54.

[3] Krutthika, H.K., & Aswatha, A.R. FPGA based design and architecture of network-on-chip router for efficient data propagation.

[4] A. Gordon-Ross, S. Abdel-Hafeez, M.H. Alsafrjalni, A One-Cycle FIFO buffer for memory management units in manycore systems, in: 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2019, pp. 265–270. IEEE.

[5] K.K. Ashok, V.K. Reddy, Advanced memory management unit for 3- D network on chip, in: 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 1062–1067. IEEE.

[6] M. LeMay, C.A. Gunter, Network-on-chip firewall: countering defective and malicious system-on-chip hardware. Logic, Rewriting, and Concurrency, Springer, Cham, 2015, pp. 404–426.

[7] Z. Yuang, L. Li, Y. Shengguang, D. Lan, L. Xiaoxiang, G. Minglun, A scalable distributed memory architecture for Network on Chip, in: APCCAS 2008-2008 IEEE Asia Pacific Conference on Circuits and Systems, 2008, pp. 1260–1263. IEEE.

[8] I. Deligiannis, G. Kornaros, Adaptive memory management scheme for MMU-less embedded systems, in: 2016 11th IEEE Symposium on Industrial Embedded Systems (SIES), 2016, pp. 1–8. IEEE.

[9] P.H. Coppock, M.K. Yacoub, B.L. Qin, A.J. Daftardar, Z. Tolaymat, V.J Mooney III, Hardware Root-of-Trust-based integrity for shared library function pointers in embedded systems, Microprocess. Microsyst. 79 (2020), 103270.

[10] A. Abbasi, J. Wetzels, T. Holz, S. Etalle, Challenges in designing exploit mitigations for deeply embedded systems, in: 2019 IEEE European Symposium on Security and Privacy (EuroS&P), 2019, pp. 31–46. IEEE.

[11] A.A. Clements, N.S. Almakhdhub, K.S. Saab, P. Srivastava, J. Koo, S. Bagchi, M. Payer, Protecting bare-metal embedded systems with privilege overlays, in: 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 289–303. IEEE.

[12] T. Abera, N. Asokan, L. Davi, J.E. Ekberg, T. Nyman, A. Paverd, G. Tsudik, C-FLAT: control-flow attestation for embedded systems software, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 743–754.

[13] M. Muench, J. Stijohann, F. Kargl, A. Francillon, D. Balzarotti, What you corrupt is not what you crash: challenges in fuzzing embedded devices, in: NDSS, 2018.

[14] M. Eceiza, J.L. Flores, M. Iturbe, Fuzzing the internet of things: a review on the techniques and challenges for efficient vulnerability discovery in embedded systems, IEEE IoT J. (2021).

[15] A. Kumar, V.K. Reddy, Advanced FIFO Structure for Router in Bi- NoC, in: 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 1219–1224. IEEE.

[16] S. Paul, N. Chatterjee, P. Ghosal, A permanent fault tolerant dynamic task allocation approach for Network-on-Chip based multicore systems, J Syst Archit 97 (2019) 287–303.

[17] A. Charif, A. Coelho, N.E. Zergainoh, M. Nicolaidis, Detailed and highly parallelizable cycle-accurate network-on-chip simulation on GPGPU, in: 2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC), 2017, pp. 672–677. IEEE.

[18] A. Firuzan, M. Modarressi, M. Daneshtalab, M. Reshadi, Reconfigurable network-on-chip for 3D neural network accelerators, in: 2018 Twelfth IEEE/ACM International Symposium on Networks-on-Chip (NOCS), 2018, pp. 1–8. IEEE.

[19] H. Jang, K. Han, S. Lee, J.J. Lee, W. Lee, MMNoC: embedding memory management units into network-on-chip for lightweight embedded systems, IEEE Access 7 (2019) 80011–80019.

[20] T. Kloda, M. Solieri, R. Mancuso, N. Capodieci, P. Valente, M. Bertogna, Deterministic memory hierarchy and virtualization for modern multi-core embedded systems, in: 2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2019, pp. 1–14. IEEE.

Debasis Behera received his B.Tech. degree in Electronics and Tele Communication engineering from the C. V. Raman College of Engineering, Bhubaneswar, India, in 2009, and the MTech. degree from National Institute of Technology, Rourkela, India, in 2011. He is continuing his Ph.D. degree in VLSI Design and Embedded System from Veer Surendra Sai University of Technology, Burla, Odisha, India. He is presently working as an Assistant professor in Electrical and Electronics Engineering department, C.V. Raman Global university, Odisha, India since 2011. His-research interests include the design of Embedded systems, Memory management, IOT, VLSI and FPGA.

Dr. Suvendu Narayan Mishra received his BE degree in Electronics and Tele communication engineering from the Orissa Engineering College, Utkal University, Odisha, India in 2003, M. Tech degree in Communication System Engineering from the Veer Surendra Sai University of Technology, Burla (VSSUT Burla), INDIA in 2010 an Ph.D. from VSSUT Burla, INDIA in 2021. He is currently working as Assistant Professor in Department of Electronics and Communication Engineering, VSSUT, Burla, INDIA. His-current research interest includes Electromagnetics, Embedded Systems, Antenna Engineering.

Dr. Prabodh Kumar Sahoo received M.E and Ph.D. degrees from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, India and Centurion University of Technology and Management, Odisha, India in 2005 and 2019 respectively. He is currently serving as Associate Professor in the department of Mechatronics, Parul University, Vadodara, Gujarat, India. His-research interest includes image processing, Computer vision, cyber physical systems.

Dr. Heli Amit Shah received her Bachelor's Degree (B. Tech.) in Electrical Engineering from Gujarat, Gujarat and Master's Degree (M. Tech.) in Automatic Control and Robotics from The Maharaja Sayajirao University of Baroda, Gujarat, India in 2003 and 2005, respectively. She has completed her Ph.D. in the field of Power Electronics, Drives and Embedded Systems from Sardar Vallabhbhai National Institute of Technology, Surat in 2015. She is currently working as Professor in Electrical Engineering Department, Parul University, Vadodara. She is inclined towards research in the field of Power Electronics, Drives and Embedded systems, since 2007, when she was working as a full-time research scholar at SVNIT, Surat. She has published more than 7 research papers in several International Journals, International and National conferences. Her current research interests include Renewable Energy Systems, Robotics and Automation, Power Electronics and Drives, IoT, and Embedded systems.